

# **BG95&BG77&BG600L Series**

## **LwM2M Application Note**

**LPWA Module Series**

Version: 1.0

Date: 2021-01-07

Status: Released



**Our aim is to provide customers with timely and comprehensive service. For any assistance, please contact our company headquarters:**

**Quectel Wireless Solutions Co., Ltd.**

Building 5, Shanghai Business Park Phase III (Area B), No.1016 Tianlin Road, Minhang District, Shanghai 200233, China

Tel: +86 21 5108 6236

Email: [info@quectel.com](mailto:info@quectel.com)

**Or our local office. For more information, please visit:**

<http://www.quectel.com/support/sales.htm>.

**For technical support, or to report documentation errors, please visit:**

<http://www.quectel.com/support/technical.htm>

Or email to [support@quectel.com](mailto:support@quectel.com).

## **General Notes**

Quectel offers the information as a service to its customers. The information provided is based upon customers' requirements. Quectel makes every effort to ensure the quality of the information it makes available. Quectel does not make any warranty as to the information contained herein, and does not accept any liability for any injury, loss or damage of any kind incurred by use of or reliance upon the information. All information supplied herein is subject to change without prior notice.

## **Disclaimer**

While Quectel has made efforts to ensure that the functions and features under development are free from errors, it is possible that these functions and features could contain errors, inaccuracies and omissions. Unless otherwise provided by valid agreement, Quectel makes no warranties of any kind, implied or express, with respect to the use of features and functions under development. To the maximum extent permitted by law, Quectel excludes all liability for any loss or damage suffered in connection with the use of the functions and features under development, regardless of whether such loss or damage may have been foreseeable.

## **Duty of Confidentiality**

The Receiving Party shall keep confidential all documentation and information provided by Quectel, except when the specific permission has been granted by Quectel. The Receiving Party shall not access or use Quectel's documentation and information for any purpose except as expressly provided herein. Furthermore, the Receiving Party shall not disclose any of the Quectel's documentation and information to any third party without the prior written consent by Quectel. For any noncompliance to the above requirements, unauthorized use, or other illegal or malicious use of the documentation and information, Quectel will reserve the right to take legal action.

## Copyright

The information contained here is proprietary technical information of Quectel. Transmitting, reproducing, disseminating and editing this document as well as using the content without permission are forbidden. Offenders will be held liable for payment of damages. All rights are reserved in the event of a patent grant or registration of a utility model or design.

*Copyright © Quectel Wireless Solutions Co., Ltd. 2021. All rights reserved.*

# About the Document

## Revision History

Version	Date	Author	Description
-	2020-12-17	Forest WANG/ Sherlock ZHAO	Creation of the document
1.0	2021-01-07	Forest WANG/ Sherlock ZHAO	First official release

## Contents

<b>About the Document</b> .....	<b>3</b>
<b>Contents</b> .....	<b>4</b>
<b>Table Index</b> .....	<b>5</b>
<b>Figure Index</b> .....	<b>6</b>
<b>1 Introduction</b> .....	<b>7</b>
1.1. Applicable Modules .....	7
<b>2 General Overview of LwM2M</b> .....	<b>9</b>
2.1. LwM2M Interfaces .....	10
2.1.1. Bootstrap Interface .....	10
2.1.2. Client Registration Interface .....	10
2.1.3. Device Management and Service Enablement Interface .....	12
2.1.4. Information Reporting Interface .....	13
2.2. LwM2M Objects .....	13
2.3. IPv4v6 Fallback Mechanism .....	14
2.4. LwM2M and PSM .....	14
2.5. File Provisioning and Configuration .....	15
2.5.1. Configuration and Script Files .....	15
2.5.2. Security Mode (DTLS) .....	22
<b>3 LwM2M AT Commands</b> .....	<b>24</b>
3.1. AT Command Syntax .....	24
3.1.1. Definitions .....	24
3.1.2. AT Command Syntax .....	24
3.2. Description of LwM2M AT Commands .....	25
3.2.1. AT+QLWCFG Configure LwM2M Client .....	25
3.2.2. AT+QLWSVC Manage LwM2M Session .....	33
3.3. Description of LwM2M URCs .....	36
3.3.1. +QLWURC: "event" LwM2M Client State Changing Notification .....	36
3.3.2. +QLWURC: "observe" Observe Request Indication .....	37
3.3.3. +QLWURC: "uldata" Application Data Delivery Result Notification .....	38
3.3.4. +QLWURC: "recv" Incoming Data Indication .....	38
<b>4 Examples</b> .....	<b>40</b>
4.1. Configure LwM2M Client with Provisioning Profiles .....	40
4.2. Configure LwM2M Client with AT Command .....	41
4.2.1. Configure in Client-Initiated Bootstrap Mode .....	41
4.2.2. Configure in Factory Bootstrap Mode .....	42
<b>5 Appendix A References</b> .....	<b>44</b>

## Table Index

Table 1: Applicable Modules .....	7
Table 2: Configuration and Script Files of LwM2M Client.....	15
Table 3: Format of bootstrap.ini .....	16
Table 4: Items of carrier_apn_cfg .....	18
Table 5: Items of lwm2m_cfg .....	19
Table 6: Types of AT Commands .....	24
Table 7: Arguments of <binding> .....	32
Table 8: Arguments of <bs_epname> and <reg_epname>.....	32
Table 9: Arguments of <state_code> and <state_string>.....	36
Table 10: Related Documents.....	44
Table 11: Terms and Abbreviations .....	44

## Figure Index

Figure 1: LwM2M Stack on the Modules.....	9
Figure 2: Operation Model for Client Initiated Bootstrap Interface .....	10
Figure 3: Client Registration Interface Example Flows.....	11
Figure 4: Example Flows of Device Management and Service Enablement Interface .....	12
Figure 5: Example Flow for Information Reporting Interface .....	13

# 1 Introduction

With the rise of the Internet of Things, more and more terminal devices have been connected to Internet. So the requirement of device management and upgrade becomes urgent.

The emergence of OMA Lightweight M2M (LwM2M) solves this industry problem. LwM2M provides some features to help device vendors to manage their devices, update the firmware, remotely control devices, etc. The main motivation for LwM2M is to define a set of lightweight protocols for a variety of IoT devices, since IoT devices are often very limited resources for embedded terminals, without UI, and come with limited computing and network communication capabilities. It is also because of the huge number of IoT terminals, due to which saving network resources becomes more important.

This document introduces LwM2M protocol, the architecture of OMA LwM2M, and how to use this feature with Quectel BG95 series, BG77 and BG600L-M3 modules.

## 1.1. Applicable Modules

Table 1: Applicable Modules

Module Series	Model	Description
<b>BG95</b>	BG95-M1	Cat M1 only
	BG95-M2	Cat M1/Cat NB2
	BG95-M3	Cat M1/Cat NB2/EGPRS
	BG95-M4	Cat M1/Cat NB2, 450 MHz Supported
	BG95-M5	Cat M1/Cat NB2/EGPRS, Power Class 3
	BG95-M6	Cat M1/Cat NB2, Power Class 3
	BG95-MF	Cat M1/Cat NB2, Wi-Fi Positioning
<b>BG77</b>	BG77	Cat M1/Cat NB2
<b>BG600L</b>	BG600L-M3	Cat M1/Cat NB2/EGPRS



**NOTE**

See the firmware release notes of corresponding module models to check whether the function has been supported.

## 2 General Overview of LwM2M

Open Mobile Alliance (OMA) specifies the application layer communication protocol between a LwM2M Server and a LwM2M Client, located in a LwM2M device. The OMA LwM2M enabler includes device management and service enablement for LwM2M devices.

The modules provide LwM2M Client on APPS. The LwM2M Client is compliant with OMA specifications and supports the following interfaces:

- Bootstrap
- Client Registration
- Device Management and Service Enablement
- Information Reporting

The following figure shows the LwM2M stack on the modules.

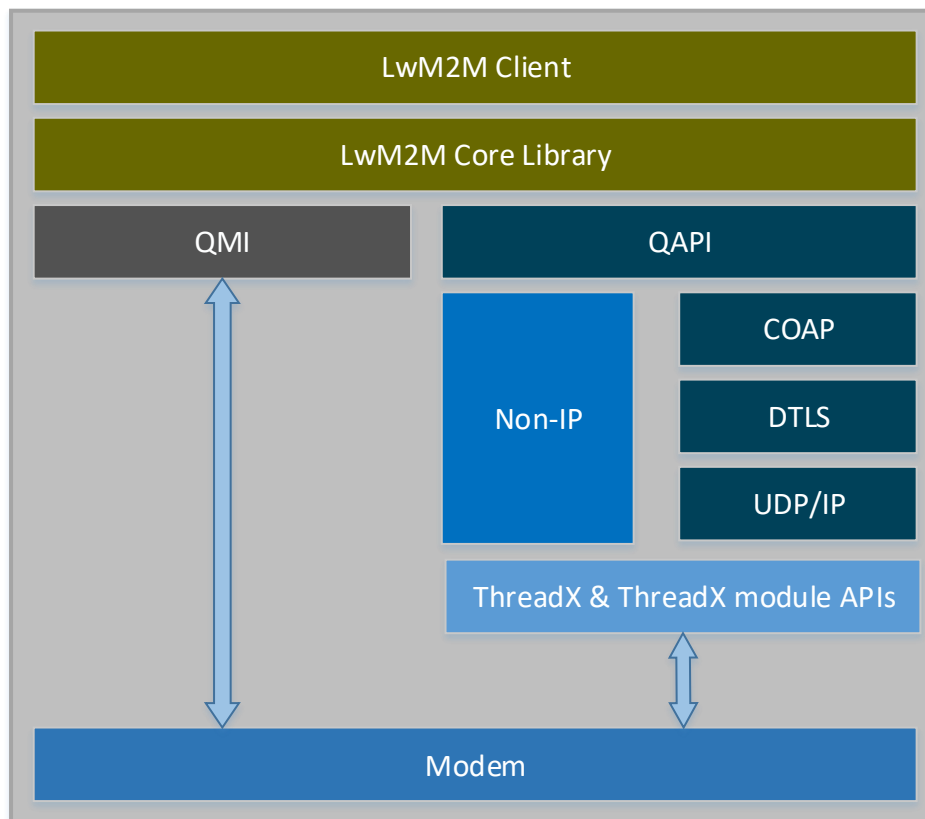


Figure 1: LwM2M Stack on the Modules

## 2.1. LwM2M Interfaces

### 2.1.1. Bootstrap Interface

The Bootstrap Interface is used to provision essential information into the LwM2M Client to enable the LwM2M Client to perform the “Register” operation with one or more LwM2M Servers.

- **Bootstrap Modes**

There are four bootstrap modes supported by the LwM2M Enabler: Factory Bootstrap, Bootstrap from Smartcard, Client Initiated Bootstrap and Server Initiated Bootstrap.

The LwM2M Client MUST support at least one bootstrap mode specified in the Bootstrap Interface. Quectel BG95 series, BG77 and BG600L-M3 modules support two bootstrap modes: Factory Bootstrap and Client Initiated Bootstrap.

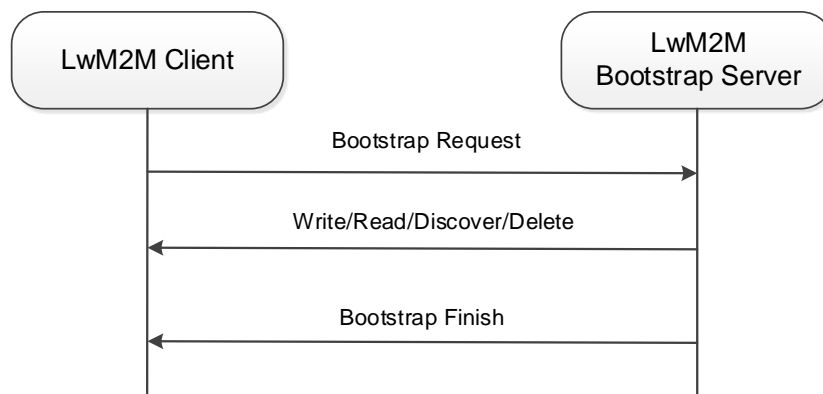


Figure 2: Operation Model for Client Initiated Bootstrap Interface

- **Server and Access Control Configurations**

Information about LwM2M bootstrap server or other standard servers, and the access right for performing an operation for these LwM2M servers.

### 2.1.2. Client Registration Interface

The Client Registration Interface is used by a LwM2M Client to register with one or more LwM2M Servers, maintain each registration, and de-register from a LwM2M Server.

- **Register**

When registering, the LwM2M Client performs the “Register” operation and provides the properties the LwM2M Server requires to contact the LwM2M Client (e.g., End Point Name); maintain the registration and session (e.g., Lifetime, Queue Mode) between the LwM2M Client and LwM2M Server as well as knowledge of the Objects the LwM2M Client supports and existing Object Instances in the LwM2M Client.

The registration is in soft state, with a lifetime indicated by the Lifetime Resource of that LwM2M Server Object Instance.

- **Update**

The LwM2M Client periodically performs an update of its registration information to the registered LwM2M Server(s) by performing the “Update” operation.

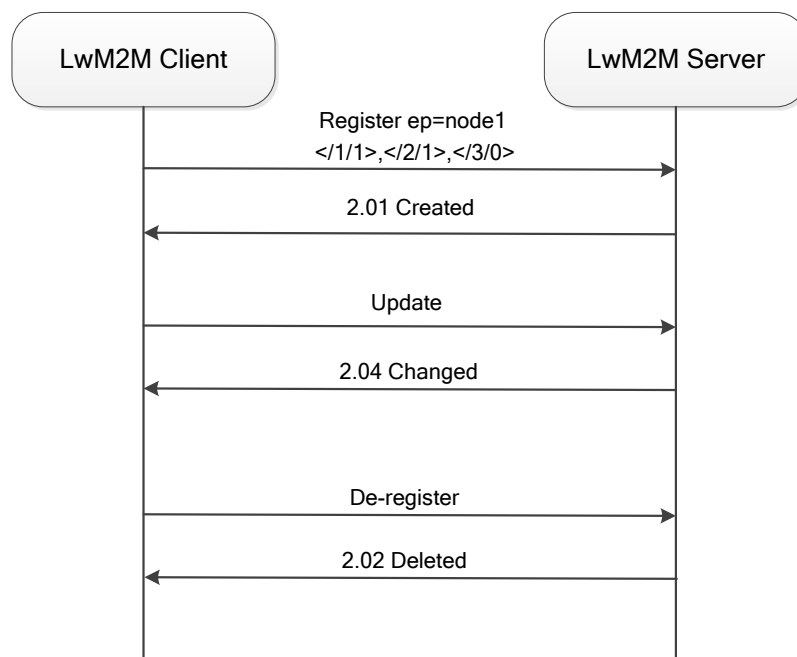
If the lifetime of a registration expires and without receiving an update from the LwM2M Client, the LwM2M Server removes the registration of that Client.

The LwM2M Client performs the “Update” operation in an interval equal to the lifetime of registration (as per Resource /1/x/1 value).

- **De-register**

Finally, when shutting down or discontinuing use of a LwM2M Server, the LwM2M Client performs a “De-register” operation.

The Binding Resource of the LwM2M Server Object informs the LwM2M Client of the transport protocol preferences of the LwM2M Server for the communication session between the LwM2M Client and LwM2M Server. The LwM2M Client SHOULD perform the operations with the modes indicated by the Binding Resource of the LwM2M Server Object Instance.



**Figure 3: Client Registration Interface Example Flows**

### 2.1.3. Device Management and Service Enablement Interface

Device Management and Service Enablement is a very important interface in LwM2M.

- The interface is used by the LwM2M Server to access Object Instances and Resources available from a registered LwM2M Client.
- The interface provides this access through the use of “Create”, “Read”, “Write”, “Delete”, “Execute”, “Write Attributes”, or “Discover” operations.
- The operations that Resource supports are defined in the Object definition using the Object Template.

The example flows of Device Management and Service Enablement Interface are shown as below:

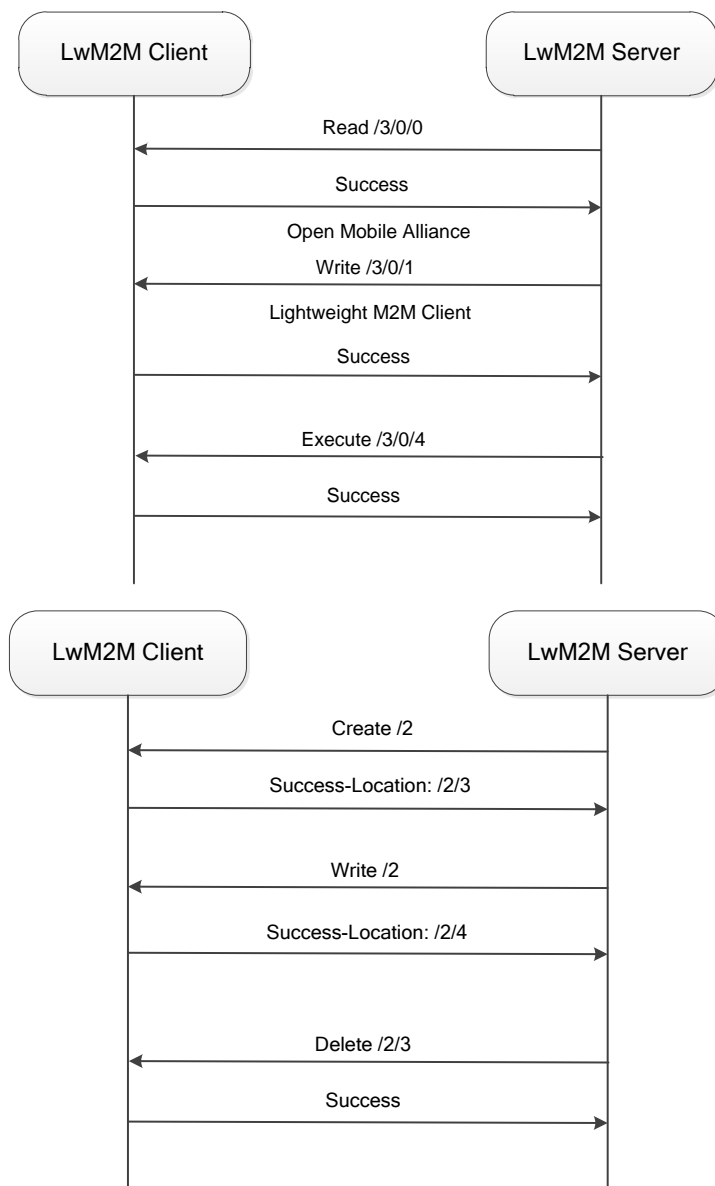


Figure 4: Example Flows of Device Management and Service Enablement Interface

### 2.1.4. Information Reporting Interface

The Information Reporting Interface is used by a LwM2M Server to observe any changes in a Resource on a registered LwM2M Client, receiving notifications when new values are available.

- This observation relationship is initiated by sending an “Observe” operation to the LwM2M Client for an Object, an Object Instance or a Resource.
- An observation ends when a “Cancel Observation” operation is performed.

LwM2M Client supports observation and notification of objects, object instances and resources.

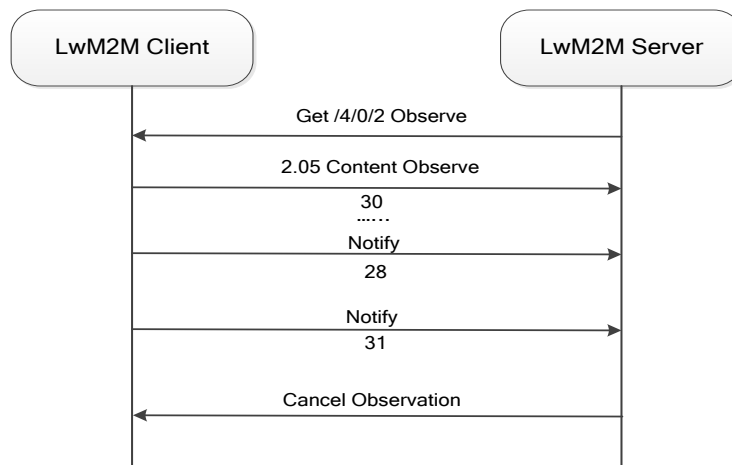


Figure 5: Example Flow for Information Reporting Interface

## 2.2. LwM2M Objects

The LwM2M Client of the modules implements the following objects and handles the server requests for them:

- Security object
- Server object
- Access control object
- Device object
- Connectivity monitoring object
- Location object
- Connectivity statistics object
- APN connection profile
- Cellular connectivity
- Firmware update object
- Software management object

- Device capability management object

The information received for these objects is sent to other registered applications, such as firmware over-the-air (FOTA), which implements these objects and monitors them using the LwM2M QAPI.

For example, the LwM2M receives a request for "execute update" on firmware update object. The LwM2M Client passes the request to the FOTA application that is registered to receive events. When the FOTA application receives the "execute update" message, it processes and applies the image and sends the result to the LwM2M Client to forward it to the LwM2M Server.

## 2.3. IPv4v6 Fallback Mechanism

If IPv4v6 is selected as the IP family:

- An IPv6 call would be brought up first.
- If the IPv6 call is not brought up even after all data call retries, then the LwM2M Client only attempts to bring up an IPv4 call.

If the module is given IPv4 and IPv6 addresses:

- The module first uses the IPv6 address to connect to the LwM2M Server.
- If the LwM2M Server is unreachable using the IPv6 address, the module falls back to the IPv4 address and tries to connect to the LwM2M Server.

## 2.4. LwM2M and PSM

- **LwM2M Client behavior**

The LwM2M client must send "registration update" to the LwM2M Server before the lifetime registration expires per the OMA specification. With the PSM enabled, the modules periodically enter into deep sleep and wake up only after the PSM timer expires. During deep sleep, the LwM2M Client is shut down and cannot send "registration update" to the server, which results in the LwM2M Client losing its registration with the server.

- **How to keep LwM2M Client registered?**

Enable the `REG_UPDATE_ON_RECONNECT` option (see **Chapter 2.5.1.3**) for the LwM2M Client to send "registration update" to the server after every wake-up from PSM. Enabling this feature ensures that the LwM2M Client does not lose its registration with the server.

**NOTE**

Currently, BG95 series, BG77 and BG600L-M3 modules only support enabling LwM2M PSM feature through provisioning profiles (see **Chapter 2.5**).

## 2.5. File Provisioning and Configuration

### 2.5.1. Configuration and Script Files

The LwM2M Client has three configuration files which can be used to control its own behavior, and one script for auto start of LwM2M Client. The four files are listed in the table below.

**Table 2: Configuration and Script Files of LwM2M Client**

Filename	Description
<i>bootstrap.ini</i>	<ul style="list-style-type: none"> <li>Used as factory bootstrap to discover the bootstrap server or other servers.</li> <li>Bootstrap server details for client initiated bootstrap.</li> <li>Can be used to instantiate multiple instances of objects (Security, Server, ACL).</li> </ul>
<i>carrier_apn_cfg</i>	Contains information of the APN for the connection to servers.
<i>lwm2m_app_autostart</i>	An empty file; the presence of this file indicates the LwM2M Client can start automatically.
<i>lwm2m_cfg</i>	Contains LwM2M options to control features such as registration retry.

**NOTE**

If the above configurations files are preloaded to the */datax* directory of the modules, the LwM2M Client starts up automatically after module reboots.

#### 2.5.1.1. bootstrap.ini

*bootstrap.ini* contains the object information required for the bootstrapping process. The object information is in JSON format defined in the OMA specification.



**Table 3: Format of bootstrap.ini**

Attributes	JSON Variable	Mandatory	Description	
Base Name	bn	No	The base name string which is prepended to the Name value of the entry for forming a globally unique identifier for resource.	
Base Time	bt	No	The base time which the Time values are relative to.	
Resource Array	e Array Parameters	Yes	The Resource list as JSON value array according to [SenML] with Array parameter extension (Object Link).	
	name	n	No	<p>The Name value is prepended by the Base Name value to form the name of the resource instance. The resulting name uniquely identifies the Resource Instance from all others.</p> <p>Example:</p> <ul style="list-style-type: none"> <li>if Base Name is "/", the Array entry Name of the Resource is {Object}/{Object Instance}/{Resource}/{Resource Instance}</li> <li>when Base Name is not present, the Array entry Name is the full URI of the requested Resource Instance</li> </ul>
	Time	t	No	The time of the representation relative to the Base Time in seconds for a notification. Required only for historical representations.
	Float Value	v	One value field is mandatory	Value as a JSON float if the Resource data type is Integer, Float, or Time.
	Boolean Value	bv		Value as a JSON Boolean if the Resource data type is Boolean.
	ObjectLink Value	ov		Value as a JSON string if the Resource data type is ObjectLink Format.(e.g."10:03")
	String Value	sv		Value as a JSON string for all other Resource data types. If the Resource data type is opaque the string value holds the Base64 encoded representation of the Resource.

The following is an example of a *bootstrap.ini* file containing security, server, and access control objects, specific to OMA 1.0 servers:

```
{
  "bn": "/0/0/",
  "e": [
    {"n": "0", "sv": "coaps://10.230.20.192:1111"},
    {"n": "1", "bv": true},
  ]
}
```

```

    {"n":"2","v":0},
    {"n":"10","v":100
  }}
{"bn":"/0/1/",
  "e":[
    {"n":"1","bv":false},
    {"n":"2","v":3},
    {"n":"10","v":102
  }}
{"bn":"/1/1/",
  "e":[
    {"n":"0","v":102},
    {"n":"1","v":50000},
    {"n":"2","v":1},
    {"n":"3","v":60},
    {"n":"5","v":86400},
    {"n":"6","bv":true},
    {"n":"7","sv":"UQS"
  }}
{"bn":"/2/0",
  "e":[
    {"n":"0","v":1},
    {"n":"1","v":2},
    {"n":"2/102","v":15},
    {"n":"3","v":102
  }}

```

The following is an example of a *bootstrap.ini* file containing security, server, and access control objects, specific to OMA 1.1 servers:

```

{"bn":"/0/0/",
  "e":[
    {"n":"0","sv":"coap://2002:c023:9c17:c007:5443:ff62:a1f5:2c:5683"},
    {"n":"1","bv":true},
    {"n":"2","v":0},
    {"n":"10","v":100
  }}
{"bn":"/0/1/",
  "e":[
    {"n":"1","bv":false},
    {"n":"2","v":3},
    {"n":"10","v":102
  }}
{"bn":"/1/1/",
  "e":[

```

```

{"n": "0", "v": 102},
{"n": "1", "v": 50000},
{"n": "2", "v": 1},
{"n": "3", "v": 60},
{"n": "5", "v": 86400},
{"n": "6", "bv": true},
{"n": "7", "sv": "UN"},
{"n": "10", "ov": "11:65533"},
{"n": "22", "sv": "U"
}}
{"bn": "/2/0",
 "e": [
  {"n": "0", "v": 1},
  {"n": "1", "v": 2},
  {"n": "2/102", "v": 15},
  {"n": "3", "v": 102
  ]}

```

### 2.5.1.2. carrier\_apn\_cfg

*carrier\_apn\_cfg* includes details about the carrier-specific APN used for each server.

**Table 4: Items of carrier\_apn\_cfg**

Parameter	Default value	Description
APN_NAME	-	APN to be used for the server with SHORT_SERVER_ID1.
APN_CLASS	2	APN class associated with the APN.
SHORT_SERVER_ID1	102	Short server ID associated with the APN.
BS_IF_REG_FAILS	0	If set to 0, LwM2M Client will not perform bootstrapping on registration failure for this SSID. If set to 1, LwM2M Client will perform bootstrapping on registration failure for this SSID.
IP_FAMILY	v4	Supported values: v4, v6 and v4v6. This is the IP family chosen to connect to the server.
BINDINGS	ip_nonip	Supported values: ip, nonip and ip_nonip. This is the binding type to be chosen to specify whether the LwM2M Client can communicate over IP or non-IP.

In order to successfully register to the network and activate the PDN connection, the corresponding APN should be configured according to the current network. After the existing APN configuration, add a semicolon and then in the next line add a new APN configuration, as shown below:

```
APN_NAME=carrier_apn1
APN_CLASS=2
BS_IF_REG_FAILS=1
IP_FAMILY=v4
BINDINGS=ip_nonip
SHORT_SERVER_ID1=102
;
APN_NAME=carrier_apn2
APN_CLASS=3
BS_IF_REG_FAILS=0
IP_FAMILY=v4
BINDINGS=ip_nonip
SHORT_SERVER_ID1=103
;
```

### 2.5.1.3. lwm2m\_cfg

*lwm2m\_cfg* is a LwM2M Client configuration file containing the following configuration information:

**Table 5: Items of lwm2m\_cfg**

Parameter	Default value	Description
APN	-	Default APN to be used if <i>carrier_apn_cfg</i> is not available.
IP_FAMILY	v4	Supported values: v4, v6, and v4v6. This is the default IP family chosen to connect to the server that does not have the IP family mentioned in <i>carrier_apn_cfg</i> .
BINDINGS	ip_nonip	Supported values: ip, nonip and ip_nonip. This is the binding type to be chosen to specify whether the LwM2M Client can communicate over IP or non-IP.
RETRY_TIMEOUT	30	Initial time period (in seconds) to retry data-call if there is a data-call failure.
RETRY_EXPONENT_VAL	2	Value by which timeout for data-call retry has to be increased exponentially.

MAX_RETRY_TIMEOUT	640	Maximum time period (in seconds) for data-call retry.
MAX_NO_RETRIES	5	Maximum number of retry attempts in case of data-call failure.
ACK_TIMEOUT	60	Sleep-ACK timeout value (in seconds). If there is no activity during the period, then the module goes to sleep.
REG_RETRY_TIMEOUT	60	Initial time period (in seconds) to retry registration, if there is a registration failure.
REG_RETRY_EXPONENT	2	Value by which timeout for registration retry must be increased exponentially.
REG_RETRY_MAXTIMEOUT	480	Maximum time period in seconds for registration retry, if there is a registration failure.
REG_UPDATE_ON_RECONNECT	0	<p>Determines whether a registration update must be sent on reconnections, such as during reboot with persistence-enabled or when IP address changes:</p> <ul style="list-style-type: none"> <li>• When set to 1, registration update is sent on reconnections.</li> <li>• When set to 0, registration update is not sent on reconnections.</li> </ul>
ROAMING_FEAT	1	<p>Determines LwM2M functionality enablement in roaming condition:</p> <ul style="list-style-type: none"> <li>• When set to 1, LwM2M functionality is disabled in roaming.</li> <li>• When set to 0, LwM2M functionality is enabled in roaming.</li> </ul>
PER_REGSTATUS_FEAT	0	<p>Determines registration persistence across reboots in LwM2M Client:</p> <ul style="list-style-type: none"> <li>• When set to 1, registration persistence is enabled across reboots in the LwM2M Client.</li> <li>• When set to 0, registration persistence is disabled across reboots in the LwM2M Client.</li> </ul>
PER_BOOTSTRAP_FEAT	1	<p>Determines the bootstrap persistence across reboots in the LwM2M client.</p> <ul style="list-style-type: none"> <li>• When set to 1, bootstrap persistence is enabled across reboots in the LwM2M client.</li> <li>• When set to 0, bootstrap persistence is disabled across reboots in the LwM2M client.</li> </ul>
BATTERY_LEVEL_THRESHOLD	20	Battery level less than the threshold mentioned in this configuration value would be considered as low battery by the client.
CARRIER_TYPE	0	<p>Enables carrier specific functionality in the client:</p> <p>0 Default</p>

		1	Verizon
REG_EP_NAME	4		EP Name to be used during registration. See OMA specification <b>Section 6.3.1 Endpoint Client Name</b> for more details. Supported values are given below: 4 IMEI URN 7 IMEI MSISDN URN
BOOTSTRAP_EP_NAME	7		EP Name to be used during bootstrap. See OMA specification <b>Section 6.3.1 Endpoint Client Name</b> for more details. Supported values are given below: 4 IMEI URN 7 IMEI MSISDN URN
BOOT_UP_SLEEP_TIME	5		Sleep time before LwM2M Client operation starts.
GPS_MIN_INTERVAL	3000		The interval (in milliseconds) after which the GNSS information is fetched periodically.
GPS_MIN_DISTANCE	1		The change in distance (in meters) after which the GNSS information is fetched.
MAX_NO_OF_FACTORY_RESET_PER_HOUR_BY_SMS	1		Number of factory-reset operations per hour allowed by SMS transport.
MAX_NO_OF_REG_UPDATE_PER_HOUR_BY_SMS	1		Number of registration update operations per hour allowed by SMS transport.
WAIT_TIME_TO_TRIGGER_NEXT_REG_CYCLE	86400		Hold-off time (in seconds) to trigger the next registration on failure of all the registration attempts.
MAX_BOOTSTRAP_WAIT_TIME	0		This is the hold-off time in seconds where bootstrap procedure is kept on hold, meanwhile extended applications can register objects.
ROOTCA_SUPPORT_MODE	0		Determines whether to use Root CA or domain issue certificate to validate certificate from the peer (bootstrap server and standard server) 0 To use domain issue certificate 3 To use Root CA
SESSION_TIMEOUT	60		Inactivity period (in seconds) after which resumption should be triggered for the uplink operation. It is recommended that OEMs configure the value to be the same as network address translation (NAT) timeout.
ENABLE_PSM	1		Determines whether LwM2M enables the use of PSM.

		0	Disable
		1	Enable
			Determines whether the device must go to PSM for every Pmin cycle.
AWAKE_AFTER_PMIN		0	0 By default, the module wakes from Pmin till Pmax.
		1	1 The module goes to PSM for every Pmin cycle till Pmax limit.
			Determines whether the LwM2M initialization is through OEM application.
STARTUP_MODE		0	0 By default, LwM2M initialization is preloaded.
		1	1 LwM2M initialization through AT commands or DAM application.
VERSION		1.0	Specifies the OMA specification version on which the LwM2M Client is running.

The following example provides an overview of the *lwm2m\_cfg* file with the options explained in the table above:

```
{
APN=carrier_apn;
RETRY_TIMEOUT=30;
RETRY_EXPONENT_VAL=2;
MAX_RETRY_TIMEOUT=640;
MAX_NO_RETRIES=5;
ACK_TIMEOUT=30;
REG_RETRY_TIMEOUT=60;
REG_RETRY_EXPONENT=2;
REG_RETRY_MAXTIMEOUT=480;
ROAMING_FEAT=1;
PER_REGSTATUS_FEAT=0;
BATTERY_LEVEL_THRESHOLD=20;
CARRIER_TYPE=0;
REG_EP_NAME=4;
BOOTSTRAP_EP_NAME=7;
BOOT_UP_SLEEP_TIME=10;
STARTUP_MODE=0;
VERSION=1.0;
}
```

### 2.5.2. Security Mode (DTLS)

Validate the following checkpoints when using pre-shared key security mode:

- PSK files corresponding to bootstrap server or standard server must be present in `/datax/ssl`.
- Files should be in the format of `<ssid>_server.psk` (e.g. `100_server.psk` for bootstrap server).
- Resource `0/x/2` decides the security mode. See **Appendix E.1** of OMA specification for possible values.

A typical PSK file used by the modules contains the following contents:

- PSK Identity used with transactions with the LwM2M Server.
- PSK Key used with transactions with the LwM2M Server.

Here is an example for credentials required for transactions with the Nokia Motive Bootstrap Server.

The Bootstrap PSK Identity structure is exactly as shown below:

```
urn:imei-msisdn:<imei_number>-<phone_number>  
Example: urn:imei-msisdn:004402460030392-2234235569
```

The PSK Key used with the bootstrap server is defined as SHA2-256 Digest of the **com.vzwm2m.com** string. As shown below:

```
PSK Key = SHA2(com.vzwm2m.com) =  
d6160c2e7c90399ee7d207a22611e3d3a87241b0462976b935341d000a91e747
```

Therefore, the content of PSK file `100_server.psk` for Nokia Motive Bootstrap Server contents are as follows:

```
[urn:imei-msisdn:004402460030392-2234235569]:  
d6160c2e7c90399ee7d207a22611e3d3a87241b0462976b935341d000a91e747
```

**NOTE**

The LwM2M Client accepts any string value as PSK Identity used with transaction, but only hexadecimal value is used as PSK Key.



# 3 LwM2M AT Commands

## 3.1. AT Command Syntax

### 3.1.1. Definitions

- **<CR>** Carriage return character.
- **<LF>** Line feed character.
- **<...>** Parameter name. Angle brackets do not appear on the command line.
- **[...]** Optional parameter of a command or an optional part of TA information response. Square brackets do not appear on the command line. When an optional parameter is not given in a command, the new value equals to its previous value or the default settings, unless otherwise specified.
- **Underline** Default setting of a parameter.

### 3.1.2. AT Command Syntax

All command lines must start with **AT** or **at** and end with **<CR>**. Information responses and result codes always start and end with a carriage return character and a line feed character: **<CR><LF><response><CR><LF>**. Throughout this document, only the commands and responses are presented, while carriage return and line feed characters are deliberately omitted.

**Table 6: Types of AT Commands**

Command Type	Syntax	Description
Test Command	<b>AT+&lt;cmd&gt;=?</b>	Test the existence of corresponding Write Command and to give information about the type, value, or range of its parameter.
Read Command	<b>AT+&lt;cmd&gt;?</b>	Check the current parameter value of a corresponding Write Command.
Write Command	<b>AT+&lt;cmd&gt;=&lt;p1&gt;[,&lt;p2&gt;[,&lt;p3&gt;[...]]]</b>	Set user-definable parameter value.
Execution Command	<b>AT+&lt;cmd&gt;</b>	Return a specific information parameter or perform a specific action.

## 3.2. Description of LwM2M AT Commands

A solution for configuring LwM2M Client with provisioning profiles is provided in **Chapter 2.5**. In this chapter, some AT commands are introduced to implement the LwM2M Client configuration.

### 3.2.1. AT+QLWCFG Configure LwM2M Client

The command configures the parameters of LwM2M Client before connecting to the LwM2M Server.

AT+QLWCFG Configure LwM2M Client	
Test Command	Response
AT+QLWCFG=?	<p>+QLWCFG: "autostart",(list of supported &lt;startup_mode&gt;s)</p> <p>+QLWCFG: "pdpcid",(range of supported &lt;client_index&gt;s),(range of supported &lt;PDP_cid&gt;s),(list of supported &lt;APN_class&gt;s)</p> <p>+QLWCFG: "security",(range of supported &lt;client_index&gt;s),(range of supported &lt;shortID&gt;s),&lt;server_addr&gt;,(list of supported &lt;bootstrap&gt;s),(list of supported &lt;security_mode&gt;s),&lt;PSK_ID&gt;,&lt;PSK_key&gt;</p> <p>+QLWCFG: "server",(range of supported &lt;client_index&gt;s),(range of supported &lt;life_time&gt;s),(range of supported &lt;period_min&gt;s),(range of supported &lt;period_max&gt;s),(range of supported &lt;disable_time&gt;s),(list of supported &lt;storing&gt;s),&lt;binding&gt;</p> <p>+QLWCFG: "epns",(list of supported &lt;type&gt;s),&lt;bs_epname&gt;,&lt;reg_epname&gt;</p> <p>+QLWCFG: "transcfg",(range of supported &lt;retry_timeout&gt;s),(range of supported &lt;retry_exponent&gt;s),(range of supported &lt;max_timeout&gt;s)</p> <p>+QLWCFG: "version",(list of supported &lt;client_version&gt;s)</p> <p>+QLWCFG: "select",(range of supported &lt;server_type&gt;s),(list of supported &lt;is_commercial&gt;s)</p> <p>+QLWCFG: "host",(list of supported &lt;host_index&gt;s),(range of supported &lt;hostID&gt;s),&lt;host_value&gt;</p> <p>+QLWCFG: "device",&lt;manufacturer&gt;,&lt;model_no&gt;,&lt;hw_version&gt;,&lt;sw_version&gt;,&lt;fw_version&gt;,&lt;device_type&gt;</p> <p>+QLWCFG: "session",(range of supported &lt;DTLS_timeout&gt;s),(range of supported &lt;resume_time&gt;s)</p> <p>+QLWCFG: "mbsps",&lt;service_code&gt;,&lt;IMEI&gt;,&lt;MSISDN&gt;,&lt;ICCID&gt;,&lt;model&gt;,&lt;MAC&gt;</p> <p>OK</p>

<p>Write Command Query/Set the startup mode of LwM2M Client. <b>AT+QLWCFG="autostart"[,&lt;startup_mode&gt;]</b></p>	<p>Response If the optional parameter is omitted, this command queries the current setting. <b>+QLWCFG: "autostart",&lt;startup_mode&gt;</b></p> <p><b>OK</b></p> <p>If the optional parameter is specified, this command sets the startup mode of LwM2M Client. <b>OK</b> Or <b>ERROR</b></p>
<p>Write Command Query/Set the APN profile for LwM2M data connection. <b>AT+QLWCFG="pdpcid"[,&lt;client_index&gt;,&lt;PDP_cid&gt;,&lt;APN_class&gt;]</b></p>	<p>Response If the optional parameters are omitted, this command queries the current setting. <b>+QLWCFG: "pdpcid",&lt;client_index&gt;,&lt;PDP_cid&gt;,&lt;APN_class&gt;</b></p> <p><b>OK</b></p> <p>If any of the optional parameters is specified, this command sets the APN profile for LwM2M data connection. <b>OK</b> Or <b>ERROR</b></p>
<p>Write Command Query/Set the resources of LwM2M security object. <b>AT+QLWCFG="security"[,&lt;client_index&gt;,&lt;shortID&gt;,&lt;server_addr&gt;,&lt;bootstrap&gt;,&lt;security_mode&gt;,&lt;PSK_ID&gt;,&lt;PSK_key&gt;]]]</b></p>	<p>Response If the optional parameters are omitted, this command queries the current setting. <b>+QLWCFG: "security",&lt;client_index&gt;,&lt;shortID&gt;,&lt;server_addr&gt;,&lt;bootstrap&gt;,&lt;security_mode&gt;</b></p> <p><b>OK</b></p> <p>If only &lt;client_index&gt; is specified and all other optional parameters are omitted, this command deletes the property data of the specified &lt;client_index&gt;. <b>OK</b></p> <p>If &lt;client_index&gt; is specified and any of the other optional parameters is specified, this command sets the specified server property. <b>OK</b> Or <b>ERROR</b></p>

<p>Write Command</p> <p>Query/Set the resources of LwM2M server object.</p> <p><b>AT+QLWCFG="server"[,&lt;client_index&gt;,&lt;life_time&gt;,&lt;period_min&gt;,&lt;period_max&gt;,&lt;disable_time&gt;,&lt;storing&gt;,&lt;binding&gt;]</b></p>	<p>Response</p> <p>If the optional parameters are omitted, this command queries the current setting.</p> <p><b>+QLWCFG: "server",&lt;client_index&gt;,&lt;life_time&gt;,&lt;period_min&gt;,&lt;period_max&gt;,&lt;disable_time&gt;,&lt;storing&gt;,&lt;binding&gt;</b></p> <p><b>OK</b></p> <p>If only &lt;client_index&gt; is specified and all other optional parameters are omitted, this command deletes the LwM2M server attributes of the specified &lt;client_index&gt;:</p> <p><b>OK</b></p> <p>If &lt;client_index&gt; is specified and any of the other optional parameters is specified, this command sets the LwM2M server attributes.</p> <p><b>OK</b></p> <p>Or</p> <p><b>ERROR</b></p>
<p>Write Command</p> <p>Query/Set the endpoint name which identifies the LwM2M Client on one LwM2M Server.</p> <p><b>AT+QLWCFG="epns"[,&lt;type&gt;,&lt;bs_epname&gt;,&lt;reg_epname&gt;]</b></p>	<p>Response</p> <p>If the optional parameters are omitted, this command queries the current setting.</p> <p><b>+QLWCFG: "epns",&lt;type&gt;,&lt;bs_epname&gt;,&lt;reg_epname&gt;</b></p> <p><b>OK</b></p> <p>If the optional parameters are specified, this command sets the endpoint name.</p> <p><b>OK</b></p> <p>Or</p> <p><b>ERROR</b></p>
<p>Write Command</p> <p>Query/Set the parameters for bootstrap or registration retry mechanism.</p> <p><b>AT+QLWCFG="transcfg"[,&lt;retry_timeout&gt;,&lt;retry_exponent&gt;,&lt;max_timeout&gt;]</b></p>	<p>Response</p> <p>If the optional parameters are omitted, this command queries the current setting.</p> <p><b>+QLWCFG: "transcfg",&lt;retry_timeout&gt;,&lt;retry_exponent&gt;,&lt;max_timeout&gt;</b></p> <p><b>OK</b></p> <p>If the optional parameters are specified, this command sets the parameters for bootstrap and registration retry mechanism.</p> <p><b>OK</b></p> <p>Or</p> <p><b>ERROR</b></p>

<p>Write Command Query/Set the version of LwM2M Client. <b>AT+QLWCFG="version"[,&lt;client_version&gt;]</b></p>	<p>Response If the optional parameter is omitted, this command queries the current setting. <b>+QLWCFG: "version",&lt;client_version&gt;</b></p> <p><b>OK</b></p> <p>If the optional parameter is specified, this command sets the version of LwM2M Client. <b>OK</b> Or <b>ERROR</b></p>
<p>Write Command Configure/query the built-in LwM2M Server information, depending on the server type specified. <b>AT+QLWCFG="select"[,&lt;server_type&gt;[,&lt;is_commercial&gt;]]</b></p>	<p>Response If the optional parameters are omitted, this command queries the current setting. <b>+QLWCFG: "select",&lt;server_type&gt;,&lt;is_commercial&gt;</b></p> <p><b>OK</b></p> <p>If any of the optional parameters is specified, this command sets the built-in LwM2M Server information. <b>OK</b> Or <b>ERROR</b></p>
<p>Write Command Query/Set the resource of portfolio object. <b>AT+QLWCFG="host",&lt;host_index&gt;[,&lt;hostID&gt;[,&lt;host_value&gt;]]</b></p>	<p>Response If the optional parameters are omitted, this command queries the current setting of the specified &lt;host_index&gt;. <b>[+QLWCFG: "host",&lt;host_index&gt;,&lt;hostID&gt;,&lt;host_value&gt;]</b> <b>[+QLWCFG: "host",&lt;host_index&gt;,&lt;hostID&gt;,&lt;host_value&gt; [...]]</b></p> <p>If only &lt;host_value&gt; is omitted, this command queries the current setting of specified &lt;host_index&gt; and &lt;hostID&gt;. <b>OK</b></p> <p>If the optional parameters are specified, this command sets the resource of portfolio object. <b>OK</b> Or <b>ERROR</b></p>
<p>Write Command Query/Set the resource of device object. <b>AT+QLWCFG="device"[,&lt;manufacturer&gt;</b></p>	<p>Response If the optional parameters are omitted, this command queries the current setting. <b>+QLWCFG: "device",&lt;manufacturer&gt;,&lt;model_no&gt;,&lt;hw_v</b></p>

<p>urer&gt;,&lt;model_no&gt;,&lt;hw_version&gt;,&lt;sw_version&gt;,&lt;fw_version&gt;,&lt;device_type&gt;]</p>	<p>ersion&gt;,&lt;sw_version&gt;,&lt;fw_version&gt;,&lt;device_type&gt;</p> <p><b>OK</b></p> <p>If the optional parameters are specified, this command sets the resource of device object.</p> <p><b>OK</b></p> <p>Or</p> <p><b>ERROR</b></p>
<p>Write Command Query/Set the session lifetime. <b>AT+QLWCFG="session" [,&lt;DTLS_timeout&gt;,&lt;resume_time&gt;]</b></p>	<p>Response</p> <p>If the optional parameters are omitted, this command queries the current setting.</p> <p><b>+QLWCFG: "session",&lt;DTLS_timeout&gt;,&lt;resume_time&gt;</b></p> <p><b>OK</b></p> <p>If the optional parameters are specified, this command sets the session lifetime.</p> <p><b>OK</b></p> <p>Or</p> <p><b>ERROR</b></p>
<p>Write Command Query/Set the value of parameters required for LG U+ LwM2M Server authentication. <b>AT+QLWCFG="mbsps" [,&lt;service_code&gt;,&lt;IMEI&gt;,&lt;MSISDN&gt;,&lt;ICCID&gt;,&lt;model&gt;,&lt;MAC&gt;]</b></p>	<p>Response</p> <p>If the optional parameters are omitted, this command queries the current setting.</p> <p><b>+QLWCFG: "mbsps",&lt;service_code&gt;,&lt;IMEI&gt;,&lt;MSISDN&gt;,&lt;ICCID&gt;,&lt;model&gt;,&lt;MAC&gt;</b></p> <p><b>OK</b></p> <p>If the optional parameters are specified, this command sets the value of parameters required for LG U+ LwM2M Server authentication.</p> <p><b>OK</b></p> <p>Or</p> <p><b>ERROR</b></p>
<p>Maximum Response Time</p>	<p>300 ms</p>
<p>Characteristics</p>	<p>The command takes effect immediately. The configurations are saved automatically.</p>

## Parameter

<p>&lt;startup_mode&gt;</p>	<p>Integer type. Enable or disable starting LwM2M automatically when the module powers on.</p>
-----------------------------	--

---

	0 Disable
	1 Enable
<client_index>	Integer type. LwM2M Client identifier. Range: 0–5.
<PDP_cid>	Integer type. PDP context ID of LwM2M Client. Range: 1–16. Default value: 1
<APN_class>	Integer type. APN class associated with the APN. The list of supported APN class are 0, 2, 3, 6, 7, 10, 11. Default value: 2.
<shortID>	Integer type. This identifier uniquely identifies each LwM2M Server configured for the LwM2M Client. Range: 1–65534.
<server_addr>	String type. LwM2M Server URI. The maximum length is 255 bytes.
<bootstrap>	Integer type. LwM2M Bootstrap Server flag. 0 Standard LwM2M Server 1 LwM2M Bootstrap Server
<security_mode>	Integer type. Encryption method for UDP payload. 0 Pre-Shared Key mode 2 Certificate mode 3 No security mode
<PSK_ID>	String type. PSK identity. Only valid when <security_mode>=0.
<PSK_key>	String type. PSK secret key. Only valid when <security_mode>=0.
<life_time>	Integer type. Lifetime of the registration. Range: 0–2147483647. Default value: 0. Unit: second.
<period_min>	Integer type. Minimum period of an Observation. Range: 0–2147483647. Default value: 0. Unit: second.
<period_max>	Integer type. Maximum period of an Observation. Range: 0–2147483647. Default value: 0. Unit: second.
<disable_time>	Integer type. The interval to the next connection after disconnecting from the LwM2M Server. Range: 60–2147483647. Default value: 86400. Unit: second.
<storing>	Integer type. The LwM2M Client stores or discards “Notify” operations to the LwM2M Server when the LwM2M Server account is disabled or the LwM2M Client is offline. 0 The LwM2M Client stores “Notify” operations 1 The LwM2M Client discards “Notify” operations
<binding>	String type. Transport binding mode configured for a LwM2M Client. See <b>Table 7</b> .
<type>	Integer type. Format of endpoint client name. 0 String type. Specify the endpoint client name with string format directly. 1 Integer type. The value is one of the enumeration specified by the module.
<bs_epname>	Endpoint client name for indentifying bootstrap server. The format is determined by <type>. See <b>Table 8</b> .
<reg_epname>	Endpoint client name for indentifying standard server. The format is determined by <type>. See <b>Table 8</b> .
<retry_timeout>	Integer type. The timeout value for each bootstrap or registration retry upon a failure. Range: 0–2147483647. Default value: 60. Unit: second.
<retry_exponent>	Integer type. Value by which the timeout for bootstrap or registration retries must be increased exponentially. Range: 0–2147483647. Default value: 2.

---

---

<b>&lt;max_timeout&gt;</b>	Integer type. The maximum timeout value for all bootstrap or registration retries upon failures. Range: 0–2147483647. Default value: 480. Unit: second.
<b>&lt;client_version&gt;</b>	Integer type. LwM2M protocol version. 0 LwM2M protocol version 1.0 1 LwM2M protocol version 1.1
<b>&lt;server_type&gt;</b>	Integer type. The built-in LwM2M Server details for network carrier. 0 Factory mode, used to reset all settings of current LwM2M Client 1 Verizon built-in LwM2M Server information 2 AT&T built-in LwM2M Server information 3 T-Mobile built-in LwM2M Server information 4 Telstra built-in LwM2M Server information 5 LG U+ built-in LwM2M Server information 6 NTT DoCoMo built-in LwM2M Server information.
<b>&lt;is_commercial&gt;</b>	Integer type. Whether the proloaded LwM2M Server details is for commercial use. 0 Staging Server 1 Production Server
<b>&lt;host_index&gt;</b>	Integer type. Instance ID of host device. Supported values are 0 and 1.
<b>&lt;hostID&gt;</b>	Integer type. Resource ID associated with host device instance. 0 Host device ID 1 Host device manufacturer 2 Host device model number 3 Host device software version
<b>&lt;host_value&gt;</b>	String type. Host device information.
<b>&lt;manufacturer&gt;</b>	String type. Device manufacturer.
<b>&lt;model_no&gt;</b>	String type. Device model number.
<b>&lt;hw_version&gt;</b>	String type. Device hardware version.
<b>&lt;sw_version&gt;</b>	String type. Device software version.
<b>&lt;fw_version&gt;</b>	String type. Device firmware version.
<b>&lt;device_type&gt;</b>	String type. Device type name.
<b>&lt;DTLS_timeout&gt;</b>	Integer type. DTLS session lifetime. Range: 0–2147483647. Default value: 0. Unit: second.
<b>&lt;resume_time&gt;</b>	Integer type. DTLS session resumption timeout. Range: 0–86400. Default value: 86400. Unit: second.
<b>&lt;service_code&gt;</b>	String type. Service code.
<b>&lt;IMEI&gt;</b>	String type. Device serial number, such as IMEI number.
<b>&lt;MSISDN&gt;</b>	String type. MSISDN number.
<b>&lt;ICCID&gt;</b>	String type. ICCID number.
<b>&lt;model&gt;</b>	String type. Device model name.
<b>&lt;MAC&gt;</b>	String type. Device MAC address.

---



**Table 7: Arguments of <binding>**

<binding>	Description
"U"	UDP. The LwM2M Server expects that the LwM2M Client is reachable via the UDP binding at any time.
"UQ"	UDP with Queue Mode. The Server MUST queue all requests to the LwM2M Client, sending requests via UDP when the LwM2M Client is on-line.
"S"	SMS. The LwM2M Server expects that the LwM2M Client is reachable via the SMS binding at any time.
"SQ"	SMS with Queue Mode. The Server MUST queue all requests to the LwM2M Client, sending requests via SMS when the LwM2M Client is on-line.
"US"	UDP and SMS. The LwM2M Server expects that the LwM2M Client is reachable via the UDP binding and SMS binding at any time.
"UQS"	UDP with Queue Mode and SMS. The Server MUST queue all requests to the LwM2M Client, sending requests via UDP when the LwM2M Client is on-line. And the LwM2M Server expects that the LwM2M Client is reachable via the SMS binding at any time.
"N"	Non-IP. The LwM2M Server MUST send requests to a LwM2M Client using the Non-IP binding. The LwM2M Client MUST send the response to such a request over the Non-IP binding.

If <type>=1, the valid values of <bs\_epname> and <reg\_epname> are listed as follows:

**Table 8: Arguments of <bs\_epname> and <reg\_epname>**

<bs_epname>/ <reg_epname>	Description
4	Identify a module using an International Mobile Equipment Identifier (see 3GPP TS 23.003). The IMEI URN specifies a valid, 15-digit IMEI. The format of the URN is "urn:imei:<IMEI>".
5	Identify a module using an Electronic Serial Number. The ESN specifies a valid, 8-digit ESN. The format of the URN is "urn:esn:<ESN>".
6	Identify a module using a Mobile Equipment Identifier. The MEID URN specifies a valid, 14-digit MEID. The format of the URN is "urn:meid:<MEID>".
7	Identify a module using a combination of International Mobile Equipment Identifier (see 3GPP TS 23.003), and MSISDN. Both IMEI and MSISDN are 15 digits. The format of the URN is "urn:imei-msisdn:<IMEI>-<MSISDN>".
8	Identify a module using a combination of International Mobile Equipment Identifier (see 3GPP TS 23.003), and IMSI. Both IMEI and IMSI are 15 digits. The format of the URN is "urn:imei-imsi:<IMEI>-<IMSI>".
9	Identify a module using a combination of International Mobile Equipment Identifier (see 3GPP TS 23.003), and IMSI. Both IMEI and IMSI are 15 digits.

The format of the URN is "imei-imsi:<IMEI>-<IMSI>".

## NOTES

1. The LwM2M security and server objects should be initialized mandatorily before LwM2M Client startup. So **AT+QLWCFG="security"** must be executed before you configure the resource values of LwM2M Server object and parameters of LwM2M data session.
2. Transport binding mode "N" is only supported under LwM2M V1.1 protocol version.

### 3.2.2. AT+QLWSVC Manage LwM2M Session

The command operates LwM2M Client and handles the behavior of LwM2M session.

AT+QLWSVC Manage LwM2M Session	
Test Command <b>AT+QLWSVC=?</b>	Response <b>+QLWSVC: "reg"</b> ,(list of supported <mode>s) <b>+QLWSVC: "dereg"</b> ,(list of supported <method>s) <b>+QLWSVC: "lifetime"</b> ,(range of supported <shortID>s),(range of supported <life_time>s) <b>+QLWSVC: "update"</b> ,(range of supported <shortID>s),(list of supported <with_object>s) <b>+QLWSVC: "state"</b> ,(range of supported <client_index>s) <b>+QLWSVC: "uldata"</b> ,(range of supported <shortID>s),<URI>,(range of supported <data_len>s),<data>,(list of supported <data_type>s),(list of supported <message_type>s)  <b>OK</b>
Write Command Initiate a registration with LwM2M Server. <b>AT+QLWSVC="reg"[,&lt;mode&gt;]</b>	Response <b>OK</b>  <b>+QLWREG: &lt;err&gt;</b>  If there is an error: <b>+CME ERROR: &lt;cme_err&gt;</b>
Write Command Deregister a LwM2M Client from the LwM2M Server. <b>AT+QLWSVC="dereg"[,&lt;method&gt;]</b>	Response <b>OK</b>  <b>+QLWDEREG: &lt;err&gt;</b>  If there is an error: <b>ERROR</b>

<p>Write Command Query/Update the lifetime of a specific LwM2M Server. <b>AT+QLWSVC="lifetime",&lt;shortID&gt;[,&lt;life_time&gt;]</b></p>	<p>Response If the optional parameter is omitted, this command queries the lifetime value of a specific LwM2M server. <b>+QLWSVC: "lifetime",&lt;shortID&gt;,&lt;life_time&gt;</b></p> <p><b>OK</b></p> <p>If the optional parameter is specified, this command updates the lifetime of a specific LwM2M server. <b>OK</b> Or <b>ERROR</b></p>
<p>Write Command Send a registration update to a specific LwM2M Server. <b>AT+QLWSVC="update",&lt;shortID&gt;[,&lt;with_object&gt;]</b></p>	<p>Response If the registration update message is sent successfully : <b>OK</b></p> <p>If there is any error: <b>ERROR</b></p>
<p>Write Command Query LwM2M Client registration state. <b>AT+QLWSVC="state"[,&lt;client_index&gt;]</b></p>	<p>Response <b>+QLWSVC: "state",&lt;client_index&gt;,&lt;shortID&gt;,&lt;server_address&gt;,&lt;bootstrap&gt;,&lt;state&gt; [...]</b></p> <p><b>OK</b></p>
<p>Write Command Send application data over LwM2M session. <b>AT+QLWSVC="uldata",&lt;shortID&gt;,&lt;URI&gt;,&lt;data_len&gt;,&lt;data&gt;[,&lt;data_type&gt;[,&lt;message_type&gt;]]</b></p>	<p>Response <b>OK</b></p> <p>If there is any error: <b>ERROR</b></p>
<p>Maximum Response Time</p>	<p>300 ms</p>
<p>Characteristics</p>	<p>/</p>

## Parameter

<p><b>&lt;mode&gt;</b></p>	<p>Integer type. Registration mode. 0 LwM2M Client initiates "registration" after the module reboots. 1 LwM2M Client initiates "registration update" after the module reboots.</p>
<p><b>&lt;method&gt;</b></p>	<p>Integer type. De-register method. 0 De-registration 1 Factory reset</p>
<p><b>&lt;shortID&gt;</b></p>	<p>Integer type. This identifier uniquely identifies each LwM2M Server configured for</p>

---

	the LwM2M Client. Range: 1–65534.
<life_time>	Integer type. The lifetime of the registration. Range: 0–86400. Unit: second.
<with_object>	Integer type. Whether the current registration update message contains an object list. <ul style="list-style-type: none"> <li>0 Send the registration update message with NULL payload.</li> <li>1 Send the registration update message with object list.</li> </ul>
<client_index>	Integer type. LwM2M Client identifier. Range: 0–5.
<server_address>	String type. LwM2M Server URI.
<bootstrap>	Integer type. LwM2M Bootstrap Server flag. <ul style="list-style-type: none"> <li>0 Standard LwM2M Server</li> <li>1 LwM2M Bootstrap Server</li> </ul>
<state>	Integer type. Registration state code of LwM2M Client. See <state_code> in <b>Chapter 3.3.1</b> .
<URI>	String type. The format of URI is /<Object ID>/<Instance ID>/<Resource ID>.
<data_len>	Integer type. Length of the application data. Range: 1–1460. Unit: byte.
<data>	String type. Application data to be sent.
<data_type>	Integer type. Type of the application data. <ul style="list-style-type: none"> <li>0 String type. Send application data in string format.</li> <li>1 Hexadecimal type. Send application data in hexadecimal format.</li> </ul>
<message_type>	Integer type. Message type of the application data to be sent. <ul style="list-style-type: none"> <li>0 Confirmable</li> <li>1 Non-confirmable</li> </ul>
<err>	Integer type. 0 indicates a successfully operation. Other values indicate a failure. <ul style="list-style-type: none"> <li>0 Operation success</li> <li>1 Operation failure</li> <li>2 Invalid parameter</li> <li>3 No memory</li> <li>4 Failure as another parallel operation started</li> <li>5 Operation is not allowed</li> </ul>
<cme_err>	Integer type. Error code of AT command operation. <ul style="list-style-type: none"> <li>701 Unknown error</li> <li>702 The command has been executed</li> <li>703 Invalid parameter</li> <li>704 LwM2M task is not initiated yet</li> <li>705 The operation is not allowed</li> <li>706 No memory</li> </ul>

---

**NOTE**

If **AT+QLWSVC="lifetime"** is executed, a "registration update" CoAP message containing lifetime value is sent to LwM2M Server. If the module fails to send this CoAP message or receives any response from LwM2M Server, the corresponding event notification in the format below is reported:

**+QLWURC: "event",<shortID>,<state\_code>,<state\_string>**

See **Chapter 3.3.1** for details of the URC.

### 3.3. Description of LwM2M URCs

The URCs of LwM2M AT commands are reported to the host when any internal event happens on LwM2M Client, including bootstrapping, registration, observation and incoming application data.

#### 3.3.1. +QLWURC: "event" LwM2M Client State Changing Notification

This URC is reported when the LwM2M Client state is changed during bootstrapping or registering; for instance, when the lifetime of a specific LwM2M server is updated, or when you send a registration update to a specific LwM2M Server.

#### +QLWURC: "event" LwM2M Client State Changing Notification

+QLWURC: "event",<shortID>,<state\_code>,<state\_string>

Report the current state of LwM2M Client.

#### Parameter

<shortID>	Integer type. Uniquely identifies each LwM2M Server configured for the LwM2M Client. Range: 0–65534. The value 0 is only used for reporting "GENERIC_FAILED" and "FACTORY_RESET" related notification.
<state_code>	Integer type. Current state code of LwM2M client. See <b>Table 9</b> .
<state_string>	String type. A readable string associated with LwM2M Client current state code. See <b>Table 9</b> .

**Table 9: Arguments of <state\_code> and <state\_string>**

<state_code>	<state_string>	Description
0	INITIAL	LwM2M Client not registered or bootstrap not started.
1	REG_PENDING	Registration pending.
2	REG_READY	Register with LwM2M Server successfully.
3	REG_FAILED	Last registration failed.
4	UPDATE_PENDING	Registration update pending due to LwM2M Client in sleep

		state.
5	FACTORY_RESET	LwM2M Client in bootstrapping state due to hold-off timer is running.
6	BS_START	LwM2M Client initiates bootstrap request.
7	BS_PENDING	Bootstrap on going.
8	BS_HOLDOFF	LwM2M client is waiting for the pass-through object initialization to complete and the "bootstrap request" message has not yet been sent.
9	BS_FINISH	Bootstrap finish.
10	BS_FAILED	Bootstrap failed.
11	REG_CONN_RETRY	Data call connection retrying on failure during registration phase.
12	REG_RETRY	Registration retrying on failure.
13	BS_CONN_RETRY	Data call connection retrying on failure during bootstrapping phase.
14	BS_RETRY	Bootstrap retrying on failure.
15	GENERIC_FAILED	Generic error which are usually casued by invalid parameters, populating endpoint name for bootstrap or registration failed and starting data call failed.
16	REG_UPDATE	Send registration update successfully.
17	UPDATE_FAILED	Send registration update failed.
18	DEREG_START	Trigger deregister from LwM2M server.
19	DEREG_FINISH	Deregister from LwM2M server successfully.

### 3.3.2. +QLWURC: "observe" Observe Request Indication

When the specific resources are observed by LwM2M Server, this URC is reported.

#### +QLWURC: "observe" Observe Request Indication

+QLWURC: "observe",<shortID><URL>,<messageID>,<token>,<flag>

Notify that there is an Observe Request from server.

## Parameter

<b>&lt;shortID&gt;</b>	Integer type. This identifier uniquely identifies each LwM2M Server configured for the LwM2M Client. Range: 1–65534.
<b>&lt;URI&gt;</b>	String type. The URI path of Observe operation. The format of URI is /<Object ID>/<Instance ID>/<Resource ID>.
<b>&lt;messageID&gt;</b>	Integer type. Message ID of Observer CoAP message.
<b>&lt;token&gt;</b>	Integer type. Token ID of Observe CoAP message.
<b>&lt;flag&gt;</b>	Integer type. Observe flag. 0 Observe 1 Cancel Observe

### 3.3.3. +QLWURC: "uldata" Application Data Delivery Result Notification

When the application data is sent in confirmable message type, this URC is reported after any response is got from the LwM2M Server.

#### +QLWURC: "uldata" Application Data Delivery Result Notification

<b>+QLWURC: "uldata",&lt;shortID&gt;,&lt;URI&gt;,&lt;messageID&gt;,&lt;token&gt;,&lt;response_code&gt;</b>	Notify the application data sending status returned from the server.
--	--

## Parameter

<b>&lt;shortID&gt;</b>	Integer type. This identifier uniquely identifies each LwM2M Server configured for the LwM2M Client. Range: 1–65534.
<b>&lt;URI&gt;</b>	String type. The format of URI is /<Object ID>/<Instance ID>/<Resource ID>.
<b>&lt;messageID&gt;</b>	Integer type. Message ID of CoAP message.
<b>&lt;token&gt;</b>	Integer type. Token ID of CoAP message.
<b>&lt;response_code&gt;</b>	Integer type. Response code used to indicate the result of sending application data. 0 Sent application data with confirmable message type successfully -1 Received error response code from server -2 Received NULL response from server -3 Received unsupported or invalid response message from server

### 3.3.4. +QLWURC: "recv" Incoming Data Indication

When the LwM2M Client has received downlink data from the server, this URC is reported.

## +QLWURC: "recv" Incoming Data Indication

+QLWURC: "recv",<URI>,<data\_type>  
<data\_len>,<CR><LF><data>

Notify incoming data in direct push mode.

### Parameter

<URI>	String type. The format of URI is /<Object ID>/<Instance ID>/<Resource ID>.
<data_type>	Integer type. Content type of incoming downlink data. 4 Resource value type is string. 5 Resource value type is opaque.
<data_len>	Integer type. Length of incoming downlink data. The maximum number of bytes that can be received at a time is 1024. Unit: byte.
<data>	The incoming downlink data.



# 4 Examples

## 4.1. Configure LwM2M Client with Provisioning Profiles

1) Use *bootstrap.ini* file to configure LwM2M security object and server object information for Leshan Server.

- Factory Bootstrap (in Non-Secure mode)

```

{"bn":"/0/1/",
 "e":[
  {"n":"0","sv":"coap://leshan.eclipseprojects.io:5683"},
  {"n":"1","bv":false},
  {"n":"2","v":3},
  {"n":"10","v":102}
 ]}
{"bn":"/1/1/",
 "e":[
  {"n":"0","v":102},
  {"n":"1","v":50000},
  {"n":"2","v":1},
  {"n":"3","v":60},
  {"n":"5","v":86400},
  {"n":"6","bv":true},
  {"n":"7","sv":"UQS"}
 ]}

```

- Factory Bootstrap (in Pre-Shared Key mode)

```

{"bn":"/0/1/",
 "e":[
  {"n":"0","sv":"coaps://leshan.eclipseprojects.io:5684"},
  {"n":"1","bv":false},
  {"n":"2","v":3},
  {"n":"10","v":102}
 ]}
{"bn":"/1/1/",
 "e":[

```

```
{ "n": "0", "v": 102 },  
{ "n": "1", "v": 50000 },  
{ "n": "2", "v": 1 },  
{ "n": "3", "v": 60 },  
{ "n": "5", "v": 86400 },  
{ "n": "6", "bv": true },  
{ "n": "7", "sv": "UQS"  
}}
```

An additional PSK key file need to be loaded into */datatx/ssl* folder in the case of Pre-Shared Key mode.

- 2) Confirm with network carrier and configure proper APN and APN class in *carrier\_apn\_cfg* and *lwm2m\_cfg* files. For details, see **Chapter 2.5.1.2** and **Chapter 2.5.1.3**.
- 3) Put the auto startup script *lwm2m\_app\_autostart* into */datatx* and reboot the module. When the module registers with Leshan Server successfully, the URCs listed below will be reported.

```
+QLWURC: "event",102,0,"INITIAL"
```

```
+QLWURC: "event",102,2,"REG_READY"
```

## 4.2. Configure LwM2M Client with AT Command

### 4.2.1. Configure in Client-Initiated Bootstrap Mode

```
//Configure LwM2M security object with pre-shared key mode.
```

```
AT+QLWCFG="security",0,111,"coaps://leshan.eclipseprojects.io:5784",1,0,"urn:imei:864475040  
090705", "313233343536373839"
```

```
OK
```

```
//Query the current resource value of LwM2M security object.
```

```
AT+QLWCFG="security"
```

```
+QLWCFG: "security",0,101,"coaps://leshan.eclipseprojects.io:5784",1,0
```

```
OK
```

```
AT+QLWCFG="server",0,60,0,50,60,1,"UQS" //Configure the LwM2M Server object.
```

```
OK
```

```
AT+QLWCFG="server"
```

```
+QLWCFG: "server",0,60,0,50,60,1,"UQS"
```

```
OK
```

```
AT+QLWSVC="reg"
```

```
//Initiate a bootstrap request to the server.
```

OK

+QLWREG: 0

+QLWURC: "event",111,6,"BS\_START"

+QLWURC: "event",111,9,"BS\_FINISH"

+QLWURC: "event",123,0,"INITIAL"

+QLWURC: "event",123,2,"REG\_READY"

//The LwM2M Client can only execute the following commands after registering with the server.

**AT+QLWSVC="lifetime",123,60** //Update the lifetime of registration to 60 s.

OK

+QLWURC: "event",123,16,"REG\_UPDATE"

**AT+QLWSVC="update",123,0** //Send "registration update" message to LwM2M Server. If the module sleeps, execute this command to wake up the module.

OK

+QLWURC: "event",123,16,"REG\_UPDATE"

//Modify "Device Object" information for the host.

**AT+QLWCFG="device","Quectel","BG95-M3","R2.1","04","BG95M3LAR02A04","Quectel\_BG95"**

OK

**AT+QLWSVC="dereg"** //LwM2M Client de-registers from LwM2M Server.

OK

+QLWURC: "event",123,19,"DEREG\_FINISH" //LwM2M Client de-registers from server successfully.

+QLWDEREG: 0

#### 4.2.2. Configure in Factory Bootstrap Mode

//Configure LwM2M security object in non-secure mode.

**AT+QLWCFG="security",0,100,"coap://leshan.eclipseprojects.io:5683",0,3**

OK

**AT+QLWCFG="server",0,60,0,50,60,1,"UQS"** //Configure LwM2M Server object.

OK

**AT+QLWSVC="reg"** //Initiate a registration request to the server.

OK

+QLWREG: 0

+QLWURC: "event",100,0,"INITIAL"

+QLWURC: "event",100,2,"REG\_READY"

//The LwM2M Client can only execute the following commands after registering with the server.

**AT+QLWSVC="lifetime",100,60**

//Update the lifetime of registration to 60 s.

OK

+QLWURC: "event",100,16,"REG\_UPDATE"

**AT+QLWSVC="update",100,0**

//Send "registration update" message to LwM2M Server, If the module sleeps, execute this command to wake up the module.

OK

+QLWURC: "event",100,16,"REG\_UPDATE"

**AT+QLWSVC="dereg",1**

//Factory reset

OK

+QLWURC: "event",0,5,"FACTORY\_RESET"

+QLWURC: "event",100,18,"DEREG\_START"

+QLWURC: "event",100,19,"DEREG\_FINISH"

+QLWDEREG: 0

# 5 Appendix A References

**Table 10: Related Documents**

SN	Document Name	Remark
[1]	OMA-TS-LightweightM2M-V1_0-20170208-A	OMA Lightweight Machine to Machine Technical Specification
[2]	OMA-AD-LightweightM2M-V1_0-20170208-A	OMA Lightweight Machine to Machine Architecture
[3]	OMA-TS-LightweightM2M_Core-V1_1_1-20190617-A	OMA Lightweight Machine to Machine Technical Specification

**Table 11: Terms and Abbreviations**

Abbreviation	Description
ACK	Acknowledgement
ACL	Access Control List
APN	Access Point Name
APPS	Application Subsystem
CoAP	Constrained Application Protocol
DAM	Downloadable Application Module
DFOTA	Delta Firmware Upgrade Over-The-Air
DTLS	Datagram Transport Layer Security
EFS	Embedded File System
EP	Endpoint
FOTA	Firmware Over-The-Air
GNSS	Global Navigation Satellite System
HTTP(S)	Hypertext Transfer Protocol (Secure)
ICCID	Integrated Circuit Card Identifier

---

IMEI	International Mobile Equipment Identity
IoT	Internet of Things
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
JSON	JavaScript Object Notation
LwM2M	Lightweight Machine to Machine
MAC	Medium Access Control
MSISDN	Mobile Subscriber Integrated Services Digital Network
NAT	Network Address Translation
OEM	Original Equipment Manufacturer
OMA	Open Mobile Alliance
PDN	Public Data Network
PDP	Packet Data Protocol
Pmax	Maximum Period
Pmin	Minimum Period
PSK	Pre-Shared Key
PSM	Power Saving Mode
QMI	Qualcomm Message Interface
SBL	Secondary Boot Loader
SenML	Sensor Measurement Lists
SMS	Short Message Service
SSID	Service Set Identifier
SSL	Secure Sockets Layer
UDP	User Datagram Protocol
UI	User Interface
URC	Unsolicited Result Code
URI	Uniform Resource Identifier
URN	Uniform Resource Name

---